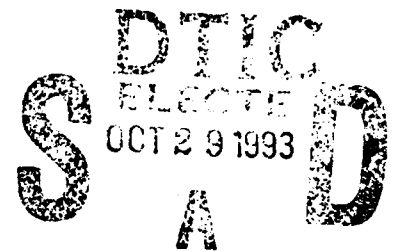


AD-A271 770



13

PI Names: Krithi Ramamritham, John A. Stankovic, Victor Yodaiken
PI Institution: University of Massachusetts
PI Phone Number: 413 545-0196 or 413 545-0720
PI E-mail Address: krithi@nirvan.cs.umass.edu, stankovic@cs.umass.edu
Grant or Contract Title: Dependable Real-Time Systems
Grant or Contract Number: N00014-85-k-0398
Reporting Period: 1 Oct 87 - 30 Sep 91



1. Productivity Measures:

Refereed papers submitted but not yet published:	12
Refereed papers published:	37
Unrefereed reports and articles:	21
Books or parts thereof submitted but not yet published:	9
Books or parts thereof published:	3
Patents filed but not yet granted:	0
Patents granted:	0
Invited presentations:	42
Contributed presentations:	52
<u>Honors received:</u>	
Technical Society Appointments	1
Conference Committee Roles	44
Editorships	4
<u>Prizes or awards received:</u>	
Promotions obtained:	1
Graduate students supported $\geq 25\%$ of full time:	14
Post-docs supported $\geq 25\%$ of full time:	1
Minorities supported (include Blacks, Hispanics, American Indians and other native Americans such as Aleuts, Pacific Islanders, etc., Asians, and Indians):	7

This document has been approved
for public release and sale; its
distribution is unlimited

93-25968



93 10 26 01

**Best
Available
Copy**

PI Names: Krithi Ramamritham, John A. Stankovic, Victor Yodaiken
 PI Institution: University of Massachusetts
 PI Phone Number: 413 545-0196 or 413 545-0720
 PI E-mail Address: krithi@nirvan.cs.umass.edu, stankovic@cs.umass.edu
 Grant or Contract Title: Dependable Real-Time Systems
 Grant or Contract Number: N00014-85-k-0398
 Reporting Period: 1 Oct 87 - 30 Sep 91

Approved for	
NIS	✓
DEC	
Director	
By	
Date	
Approved	
Date	Approved
A-J	

2. Summary of Accomplishments

2.1 Our Approach and its Advantages

Most current real-time operating systems contain the same basic paradigms found in timesharing operating systems and are simply stripped down and optimized versions of timesharing operating systems. For example, while they stress fast mechanisms such as a fast context switch and the ability to respond to external interrupts quickly, they retain the main abstractions of timesharing operating systems. In addition, very often today's real-time kernels use priority scheduling. Priority scheduling is a mechanism which provides no direct support for meeting timing constraints. For example, the current technology burdens the designer with the unenviable task of mapping a set of specified constraints on task executions into task priorities in such a manner that all tasks will meet their deadlines. Thus, when using the current paradigms together with priority scheduling it is difficult to *predict* how tasks, dynamically invoked, interact with other active tasks, where blocking over resources will occur, and what the subsequent effect of this interaction and blocking is on the timing constraints of all the tasks. Basically, currently used scheduling policies are inadequate for three main reasons: (1) they do not address the need for an integrated cpu scheduling and resource allocation scheme, (2) they don't handle the end-to-end scheduling problem, and (3) they are not used in a planning mode, thereby containing a myopic view of the system capabilities.

Because of these reasons, even though some commercial and research operating systems have bounded execution times and hence are predictable in a low-level sense, they do not provide the techniques or tools to support application level predictability. This is what our approach attempts to do.

Specifically, our approach is based on categorizing the types of tasks that occur in a real-time system and then providing the necessary support for them in an integrated fashion. Tasks found in real-time applications can be categorized on the basis of their interaction with and impact on the environment. This gives rise to two main criteria: importance and timing requirements. This leads to the identification of three types of tasks: critical tasks, essential tasks, and non-essential tasks. Our kernel then treats the different classes

of tasks differently thereby reducing the overall complexity. The technical underpinnings of the Spring paradigm can be summarized as follows:

- Tasks are part of a single application with a system-wide objective. The types of tasks that occur in a real-time application are known *a priori* and hence can be analyzed to determine their characteristics (such as their importance, as well as their timing and resource requirements).

The scheduling and allocation schemes adopted in Spring use this information in pre-allocation and for on-line guarantee of timing constraints.

- Predictability should be ensured so that the timing properties of both individual tasks and the system can be assessed (in other words we have to be able to categorize the performance of tasks and the system with respect to properties such as timing and fault tolerance).

In Spring, *predictability* is achieved by a combination of schemes, including resource segmentation/partitioning, functional partitioning of application tasks, executing system support tasks on a separate processor, and the use of integrated scheduling algorithms.

- Flexibility should be ensured so that system modifications and on-line dynamics are more easily accommodated.

Flexibility/adaptability is improved in Spring by dynamic (decentralized) task scheduling, and the use of meta-level control.

- The value of tasks executed should be maximized, where the value of a task that completes before its deadline is its full value (depends on what the task does) and some diminished value (e.g., a very negative value or zero) if it does not make its deadline. Fairness and minimizing average response times are not important metrics for tasks with hard timing constraints. We should note that while some of the work at CMU captures task values through (arbitrary) value functions, for efficiency reasons, we capture values through task categorization and importance factors. In addition we explicitly account for resource requirements of tasks.

Spring maximizes the *value* of tasks executed through resource preallocation for critical tasks and the use of dynamic scheduling algorithms (that take task importance values into account) for essential and non-essential tasks.

An important aspect of our paradigm is that the system maintains significant semantic information concerning the application in on-line data structures. Consequently, task group information, resource requirements, importance, timing constraints, and fault tolerance requirements are available to the system algorithms such as the dynamic guarantee algorithm or recovery algorithms. These algorithms can then make use of this information to directly support the application.

It is important to note that the system structures and algorithms in the kernel are generic; it is just the specific task information which is in these data structures that is application

dependent. However, we are very interested in showing how the generic kernel can be used in domain specific applications such as process control, avionics, mobile robotics and command and control.

2.2 Specifics of the Accomplishments

Based on the paradigm provided in the previous section, we (1) made great strides in developing a sound approach to scheduling tasks in complex real-time systems, (2) developed a real-time operating system kernel, a preliminary version of which is in operation, and (3) we have developed a modal primitive recursive arithmetic for specifying and verifying real-time systems.

To study different scheduling approaches and to evaluate different real-time architectures, we have developed a simulation testbed with a graphical front-end that allows a system developer to select the task management approaches appropriate for a given application. Using such a testbed, the efficacy of the scheduling algorithms has been carefully demonstrated via extensive simulation studies. Also, we have conducted mathematical analysis of the algorithms to understand their worst-case behavior. Finally, and most importantly, some of the algorithms have also been implemented as part of the kernel.

We now provide a brief summary of our work on the scheduling algorithms, on the Spring kernel, and on formal aspects of real-time systems.

2.2.1 Scheduling Strategies for Complex Real-Time Systems

To date, the main results of our scheduling work include:

- *The adoption and development of guarantee, a notion fundamental to predictable scheduling.* A task is guaranteed by constructing a plan for task execution whereby all guaranteed tasks meet their timing constraints. A task is guaranteed subject to a set of assumptions, for example, about its worst case execution time, and the nature of faults in the system. If these assumptions hold, once a task is guaranteed it *will* meet its timing requirements.
- *Algorithms for the guarantee of dynamically arriving tasks.* Our guarantee algorithm has the potential to deal with tasks that have deadlines, resource requirements, precedence relationships among tasks, tasks with varying importance levels, tasks with different fault tolerance requirements, and tasks that are preemptable. Many versions of the algorithms exist each dealing with a subset of these task characteristics. As mentioned earlier, one of our goals is to produce comprehensive schemes that can deal with tasks having all these complex characteristics.
- *Analysis of the quality of the schedules produced by the algorithm.* Both the ability to generate feasible schedules and the *quality* of the generated feasible schedules, expressed

in terms of the schedule length, are important metrics for scheduling algorithms. Our theoretical analysis identified several ways in which the basic algorithm can be improved even further.

- *Reclaiming unused time and resources when tasks complete early.* The guarantees are based on worst case computation times, but when a task finishes early, as would typically be the case, the unused CPU and resource time may be reclaimed and used.
- *Distributed scheduling and meta-level control.* A suite of distributed scheduling algorithms has been developed and evaluated. We have also hypothesized the usefulness of a *meta-level controller* that can select the heuristic appropriate for a given system (state). We have developed a local area network architecture for communication in real-time systems that incorporates the abstractions of real-time datagrams and real-time virtual circuits. We have also developed several real-time communication protocols for multiple-access channels. We have yet to integrate these protocols with the distributed scheduling algorithms.
- *Static allocation and scheduling of safety-critical tasks.* We have developed an algorithm that is suitable for the static allocation and scheduling of complex, safety-critical periodic tasks. Besides periodicity constraints, tasks handled by the algorithm can have resource requirements and can possess precedence, communication, as well as fault tolerance constraints. Extensions necessary to accommodate dynamic aperiodic arrivals have also been worked out.

The notion of guarantee is aimed at achieving *predictability*. *Adaptability* and *flexibility* are supported by a collection of dynamic task management techniques. These include dynamic guarantees with task characteristics identified at invocation time, distributed scheduling, meta-level control, and resource reclaiming.

It is important to mention that we have also been able to apply our ideas from real-time scheduling algorithms to develop scheduling and concurrency control techniques for real-time database systems. This work has produced significant results. It is also important to point out that this work represents the first set of results for a real-time database in an experimental testbed environment. All other results we are aware of in this area are based on simulation studies.

2.2.2 The Spring Real-Time Operating System Kernel

The Spring kernel stresses the real-time and flexibility requirements, and also contains several features to support fault tolerance. Our approach to supporting this new paradigm combines the following ideas resulting, we believe, in a flexible yet predictable system:

- resource segmentation/partitioning,
- functional partitioning,
- selective preallocation,
- *a priori* guarantee for critical tasks,

- an on-line guarantee for essential tasks,
- integrated cpu scheduling and resource allocation,
- use of the scheduler in a planning mode,
- the separation of importance and timing constraints, e.g., a deadline,
- end-to-end scheduling, and
- the utilization of information about tasks at *run time* including timing, task importance, fault tolerance requirements, etc. and the ability to dynamically alter this information.

Based on these principles, we are currently developing the kernel for distributed real-time system. The current configuration is composed of a network of multiprocessors each running the Spring kernel. Each multiprocessor contains one (or more) application processors, one (or more) system processors, and an I/O subsystem. Application processors execute relatively high level application tasks which have been previously guaranteed. System processors offload the scheduling algorithm and other OS overhead from the application tasks both for speed, and so that external interrupts and OS overhead do not cause uncertainty in executing guaranteed tasks. The I/O subsystem is partitioned away from the Spring kernel and it handles I/O and interactions with sensors.

Not surprisingly, the main components of the kernel can be grouped into task management and scheduling, memory management, and intertask communication. While this sounds similar to many other kernels, the abstractions supported are quite different and represent a new paradigm for real-time operating systems. To enhance predictability, system primitives have capped execution times, and some primitives execute as iterative algorithms where the number of iterations it will make for a particular call depends on its capped execution time and on other state information including available time.

Nodes in the Spring distributed system are connected via Ethernet (for non-real-time traffic) as well as via a predictable replicated memory based on SCRAMNET (developed by SYSTRAN) - a fiber optic register insertion ring (for real-time traffic). The development of software required to support SCRAMNET is currently underway.

2.2.3 Formal Approaches to Real-Time Systems

In the area of formal approaches to real-time systems, our work has attempted to find new techniques for describing real-time systems as finite state machines. We have developed methods of specifying the behavior and structure of large-scale state machines symbolically, without enumerating states.

We use state-dependent integer functions that implicitly reference a state machine. Each boolean function specifies the family of state machines which cause it to evaluate as true. This approach allows for very compact, parameterizable specifications of state machines which are beyond the reach of traditional methods. We can conveniently compose these functions to specify the behavior of systems composed from concurrent subsystems. Composite specifications also specify finite state machines and can be used as parts of more complex specifications. Our composition method has a particular advantage over other methods in

that it does not require any assumptions about how components are interconnected or scheduled and the composite specification also specifies a finite state machine. Specifications are just integer-valued functions, thus verification takes place in the familiar context of integer mathematics.

The compactness of our specifications of state machines allows us to model the behavior of real-time systems with state machines that change state in response to the passage of physical time, e.g., every nano-second. The flexibility of our composition method allows us to describe real-time systems containing subsystems which change state at differing rates, and which may contain clocks that drift apart at different rates.

We have applied our formalism to specify real-time priority queues, a fault-tolerant broadcast protocol, and the arbitration mechanism of Futurebus+.

PI Names: Krithi Ramamritham, John A. Stankovic, Victor Yodaiken
PI Institution: University of Massachusetts
PI Phone Number: 413 545-0196 or 413 545-0720
PI E-mail Address: krithi@nirvan.cs.umass.edu, stankovic@cs.umass.edu
Grant or Contract Title: Dependable Real-Time Systems
Grant or Contract Number: N00014-85-k-0398
Reporting Period: 1 Oct 87 - 30 Sep 91

3. Summary

For the most part, we accomplished what we outlined in the proposal submitted at the beginning of the ONR initiative. Many versions of the dynamic and static scheduling problems have been studied and solutions have been evaluated and in some cases even implemented on the Spring kernel. In addition, a few new and interesting problems came to our attention as a result of our endeavor to implement our scheduling algorithm on the kernel. These include the resource reclaiming problem and the problem of predictable and bounded synchronization in a shared-bus multiprocessor system. In addition, our work on formal aspects of real-time systems has matured through its application to many nontrivial examples.

PI Names: Krithi Ramamritham, John A. Stankovic, Victor Yodaiken
PI Institution: University of Massachusetts
PI Phone Number: 413 545-0196 or 413 545-0720
PI E-mail Address: krithi@nirvan.cs.umass.edu, stankovic@cs.umass.edu
Grant or Contract Title: Dependable Real-Time Systems
Grant or Contract Number: N00014-85-k-0398
Reporting Period: 1 Oct 87 - 30 Sep 91

4. Publications, Presentations, and Reports:

4.1 Publications and Reports:

1. Arvind, K., "A New Probabilistic Algorithm for Clock Synchronization," *Proc. Real-Time Systems Symposium*, December 1989. Also submitted to *IEEE Transactions on Parallel and Distributed Systems*, Jan. 1990.
2. Arvind, K., K. Ramamritham, and J. A. Stankovic, "Integrated Protocols for Real-Time Virtual Circuit and Real-Time Datagram Services," submitted to *11th Real-time Symposium*.
3. Arvind, K., K. Ramamritham, J. A. Stankovic, "Window MAC Protocols for Real-Time Communication Services," submitted to *IEEE Transactions on Communication*, Feb. 1991. Also COINS Technical Report 90-127.
4. Arvind, K., K. Ramamritham and J. A. Stankovic, "A Local Area Network Architecture for Communication in Distributed Real-Time Systems," invited paper, *Real-Time Systems Journal*, Vol. 3, No. 2, pp. 113-147, May 1991.
5. Baruah, S., G. Koren, D. Mao, B. Mishra, L. Rosier, D. Shasha and F. Wang, "On the Competitiveness of On-Line Real-Time Task Scheduling," to appear, *Proc. Real-Time Systems Symposium*, Dec 1991.
6. Biyabani, S., J. A. Stankovic, and K. Ramamritham, "The Integration of Criticalness and Deadline In Scheduling Hard Real-Time Tasks," *Real-Time Systems Symposium*, December 1988, Also COINS Technical Report 88-82.
7. Chen, S., J. A. Stankovic, J. Kurose and D. Towsley, "Performance Evaluation of Two New Disk Scheduling Algorithms for Real-Time Systems," to appear, *Real-Time Systems Journal*.
8. Gehani, N. and K. Ramamritham, "Real-Time Concurrent C (C++): A Language for Programming Dynamic Real-time Systems," (to appear) *Real-Time Systems Journal*, Vol 4, No 3, 1991.

9. Huang, J. and J. A. Stankovic, "Real-Time Buffer Management," submitted to *ACM Sigmetrics*, September 1990.
10. Huang, J., J. A. Stankovic, K. Ramamritham and D. Towsley, "Experimental Evaluation of Real-Time Optimistic Concurrency Control Schemes," *Conference on Very Large Data Bases*, Sep 1991.
11. Huang, J., J. A. Stankovic, K. Ramamritham and D. Towsley, "On Using Priority Inheritance in Real-Time Databases," to appear, *Real-Time Systems Symposium*, Dec 1991.
12. Huang, J., J. A. Stankovic, D. Towsley, and K. Ramamritham, "Experimental Evaluation of Real-Time Transaction Processing," *Real-Time Systems Symposium*, pp. 144-153, Dec 1989.
13. Huang, J., J. A. Stankovic, D. Towsley and K. Ramamritham, "Real-Time Transaction processing: Design, Implementation, and Performance Evaluation," submitted to *IEEE Trans. on Knowledge and Database Engineering*, May 1990, under revision.
14. Mirchandaney, R., D Towsley, and J. Stankovic, "Analysis of the Effects of Delays on Load Sharing," accepted for publication in *IEEE Transactions on Computers*.
15. Molesky, L., C. Shen, and G. Zlokapa, "Predictable Synchronization Mechanisms for Multiprocessor Real-Time Systems," *Real-Time Systems Journal*, Sept. 1990. Also COINS Technical Report 89-106.
16. Molesky, L., K. Ramamritham, C. Shen, J. A. Stankovic, and G. Zlokapa, "Implementing a Predictable Real-Time Multiprocessor Kernel - The Spring Kernel," extended abstract, *Proc. IEEE Workshop on Real-Time Operating Systems and Software*, May 1990. Also appeared in *Real-Time Systems Newsletter*, Vol. 6, No. 2, Spring 1990.
17. Niehaus, D., "Program Representation and Translation to Support Predictability of Real-Time Systems," to appear, *Proceedings of the IEEE Real-Time Systems Symposium*, 1991.
18. Niehaus, D., E. Nahum and J. A. Stankovic, "Predictable Real-Time Caching in the Spring System," extended abstract, *Proceedings of the Joint IEEE Workshop on Real-Time Operating Systems and Software, and IFAC Workshop on Real-Time Programming*, pp. 80-87, 1991.
19. Niehaus, D., J. A. Stankovic, and K. Ramamritham, "System Support for Real-Time AI: A Spring Project Perspective," *Workshop on Real-Time AI, IJCAI*, August 1989.
20. Rajkumar, R., L. Sha, J. P. Lehoczky, and K. Ramamritham, "An Optimal Priority Inheritance Protocol for Real-Time Synchronization," (revised for second round of reviews) *ACM Transactions on Programming Language and Systems*, April 1991.

21. Ramamritham, K., "Work at Eindhoven on Formal Aspects of Real-time Systems," *ONR European Scientific News*, 1988.
22. Ramamritham, K., "The MARS Project: Building Deterministic Real-time Systems," *ONR European Scientific News*, 1989.
23. Ramamritham, K., "Assignment and Scheduling of Complex Periodic Tasks," (to be submitted to) *International Conference on Distributed Computing Systems*, Oct. 1989.
24. Ramamritham, K., "Time Constrained Object-Oriented Computations," Technical Memo, University of Newcastle, June 1988.
25. Ramamritham, K., "Allocation and Scheduling of Complex Periodic Tasks," *10th International Conference on Distributed Computing Systems*, Paris, France, June 1990. Submitted to *IEEE Transactions on Parallel and Distributed Systems*.
26. Ramamritham, K., "Fault Tolerance in Spring: Issues and Possibilities," COINS Technical Report, Oct. 1990.
27. Ramamritham, K., "Scheduling Complex Periodic Tasks," (submitted to) *IEEE Transactions on Parallel and Distributed Systems*, 1990.
28. Ramamritham, K., and J. A. Coello, "Load Balancing During the Static Allocation and Scheduling of Complex Periodic Tasks," (submitted for publication, Oct. 1990).
29. Ramamritham, K., and N. Gehani, "Real-Time Concurrent C (C++): A Language for Programming Dynamic Real-Time Systems," Oct. 1990.
30. Ramamritham, K. and J. Stankovic, "Time-Constrained Communication Protocols for Hard Real-Time Systems," extended abstract, *IEEE Workshop on Real-Time Operating Systems*, May 1989.
31. Ramamritham, K. and J. A. Stankovic, "Scheduling Results in the Spring Project: An Overview," chapter in Foundations of Real-Time Computing: Scheduling and Resource Management, edited by Andre van Tilborg and Gary Koob, Kluwer Academic Publishers, pp. 277-306, 1991.
32. Ramamritham, K., and J. A. Stankovic, "Scheduling Strategies Adopted in Spring: An Overview," to appear Foundations of Real-Time Computing: Resource Management and Formal Specifications, Andre van Tilborg, Editor, Office of Naval Research.
33. Ramamritham, K., J. A. Stankovic, and P. Shieh, "Efficient Scheduling Algorithms for Real-Time Multiprocessor Systems," *IEEE Transactions on Parallel and Distributed Computing*, Vol. 1, No. 2, pp. 184-194, April 1990.
34. Ramamritham, K., J. A. Stankovic, and W. Zhao, "Distributed Scheduling of Tasks With Deadlines and Resource Requirements," *IEEE Transactions on Computers*, Vol. 38, No. 8, August 1989, pp. 1110-1123.

35. Ramamritham, K., J. A. Stankovic, and P. Shiah, "O(n) Scheduling Algorithms for Real-Time Multiprocessor Systems," *Proc. Int. Conf. on Parallel Processing*, August 1989.
36. Ramamritham, K., J. A. Stankovic, and P. Shiah, "Efficient Scheduling Algorithms for Real-Time Multiprocessor Systems," (submitted to) *IEEE Transactions on Parallel and Distributed Systems*, March 1989.
37. Sha, L., R. Rajkumar, J. Lehoczky, and K. Ramamritham, "Mode Change Protocols for Priority-Driven Preemptive Scheduling," *Real-Time Systems*, pp. 243-264, Vol. 1, No. 3, Dec. 1989.
38. Shen, C., K. Ramamritham, J. A. Stankovic, "Resource Reclaiming in Multiprocessor Real-Time Systems," (submitted to) *IEEE Transactions on Parallel and Distributed Systems*, Feb. 1991. Also *Proceedings Real-Time Systems Symposium*, Dec. 1990.
39. Shen, C., K. Ramamritham, and J. A. Stankovic, "The Spring Project: Integrated Resource Management in Real-Time Systems," *Proc. Third Workshop on Large Grain Parallelism*, Oct. 1989.
40. Shih, C. and J. A. Stankovic, "Deadlock Detection in Distributed Concurrent Programming Environments and Its Application to Real-Time Systems and ADA," invited paper, *Journal of Computer Science and Informatics*, Computer Society of India, to appear.
41. Shih, C. and J. A. Stankovic, "Distributed Deadlock Detection in Ada Runtime Environments," *TRI-ADA 90*, October 1990.
42. Shih, C. and J. A. Stankovic, "Survey of Deadlock Detection in Distributed Concurrent programming Environments and Its Application to Real-Time Systems," COINS TR90-69, June 1990.
43. Stankovic, J. A. "Decentralized Decision Making For Task Reallocation In a Hard Real-Time System," *IEEE Transactions on Computers*, Vol. 38, No. 3, March 1989, pp. 341-355.
44. Stankovic, J. A., "Real-Time Computing," Chapter in *Encyclopedia of Computer Science and Technology*, Marcel Dekker, Inc., NY, NY, accepted for publication.
45. Stankovic, J. A., "The Spring Architecture," *Proc. EuroMicro Workshop on Real-Time*, Denmark, June 1990.
46. Stankovic, J. A., "Misconceptions About Real-Time Computing: A Serious Problem for Next Generation Systems," *IEEE Computer*, Vol. 21, No. 10, Oct. 1988.
47. Stankovic, J. A., "On the Reflective Nature of the Spring Kernel," *invited paper*, to appear *Proc. Process Control Systems '91*, Feb. 1991.

48. Stankovic, J. A., "Distributed Computing," invited paper, The Encyclopedia of Telecommunications, Marcel Dekker, Inc., NY, NY, in preparation.
49. Stankovic, J. A. and K. Ramamritham, "The Design of the Spring Kernel," *Proceedings 1987 Real-Time Systems Symposium*, Dec 1987.
50. Stankovic, J. A. and K. Ramamritham, Hard Real-Time Systems, Tutorial Text, IEEE Computer Society Press, Washington DC 1988, 618 pages.
51. Stankovic, J. A. and K. Ramamritham, Advances in Hard Real-Time Systems, Tutorial Text, IEEE Computer Society Press, in preparation, proposal accepted July 1991.
52. Stankovic, J. A. and K. Ramamritham, "The Spring Kernel: A New Paradigm for Real-Time Operating Systems," *ACM Operating Systems Review*, Vol. 23, No. 3, July, 1989, pp. 54-71.
53. Stankovic, J. A. and K. Ramamritham, "The Spring Kernel: A New Paradigm for Real-Time Operating Systems," extended abstract, *IEEE Workshop on Real-Time Operating Systems*, May 1989.
54. Stankovic, J. A. and K. Ramamritham, "The Spring Kernel: Support for Next Generation Real-Time Operating Systems," submitted to *IEEE Computer*, Aug. 1990.
55. Stankovic, J. A. and K. Ramamritham, "What is Predictability for Real-Time Systems - An Editorial," *Real-Time Systems Journal*, Vol. 2, pp. 247-54, Dec. 1990.
56. Stankovic, J. A. and K. Ramamritham, "The Spring Kernel: Operating System Support for Critical, Hard Real-Time Systems," Chapter in Operating Systems for Mission Critical Computing, edited by K. Gordon, P. Hwang and A. Agrawala, to appear ACM Press.
57. Stankovic, J. A. and K. Ramamritham "The Spring Kernel: A New Paradigm for Hard Real-time Operating Systems," *IEEE Software*, Vol. 8, No. 3, pp. 62-72, May 1991; (to appear) in *Operating Systems for Mission Critical Computing*, Agrawala and Gordon, and Hwang, editors. Also *ACM Operating Systems Review*, Vol. 23 No. 3, July 89, pp. 54-71. Extended abstract *IEEE Workshop on RTOS*, May 1989.
58. Stankovic, J. A., K. Ramamritham and E. M. Nahum, "Predictable Interprocess Communication for Hard Real-Time Systems," *10th IFAC Workshop on Distributed Computer Control Systems*, Sept. 1991.
59. Stankovic, J., K. Ramamritham, D. Niehaus, "On Using the Spring Kernel to Support Real-Time AI Applications," *Proc. EuroMicro Workshop on Real-Time Systems*, June 1989.

60. Stankovic, J. A., K. Ramamritham, and D. Towsley, "Scheduling in Real-Time Transactions," chapter in Foundations of Real-Time Computing: Scheduling and Resource Management, edited by Andre van Tilborg and Gary Koob, Kluwer Academic Publishers, pp. 157-184, 1991.
61. Stankovic, J. A., K. Ramamritham, C. Shieh, and W. Zhao, "Real-Time Scheduling Algorithms for Multiprocessors," COINS TR 89-47.
62. Stankovic, J. A. and W. Zhao, "On Real-Time Transactions," *ACM SIGMOD Record*, Special Issue on Real-Time Database Systems, Vol. 17, No. 1, pp. 4-18, March 1988.
63. Towsley, D., G. Rommel, and J. A. Stankovic, "Analysis of Fork-Join Program Response Times on Multiprocessors," accepted for publication in *IEEE Transactions on Computers*.
64. Wang, F. and M. Decao, "Worst Case Analysis for On-Line Scheduling in Real-Time Systems," COINS TR91-54.
65. Wang, F., K. Ramamritham and J. A. Stankovic, "Bounds on the Schedule Length for Some Heuristic Scheduling Algorithms for Hard Real-Time Tasks," (submitted to) *the Journal of the ACM* publication, August 1990.
66. Yodaiken, V., "Regular Languages and Products of Regular Languages as Kripke Semantics," COINS Technical Report 89-58.
67. Yodaiken, V., "Finite State Modal Arithmetic: A New Approach to Finite Automata."
68. Yodaiken, V., "The Algebraic Feedback Product of Automata," WAV 1990 (Springer-Verlag LNCS series and longer version in ACM/AMS Series).
69. V. Yodaiken, "A Logic-Free Method for Compositional Specification," Proceedings, ICCI 1990 (Springer-Verlag LNCS series).
70. Yodaiken, V., and K. Ramamritham, "An Arithmetic for Real-Time Systems," to appear Foundations of Real-Time Computing: Resource Management and Formal Specifications, Andre van Tilborg, Editor, Office of Naval Research.
71. Yodaiken, V., K. Ramamritham, "Specifying and Verifying a Real-Time Priority Queue with Modal Algebra," To appear *Proceedings Real-Time Systems Symposium*, Dec. 1990.
72. Yodaiken, V. and K. Ramamritham, "Hierarchical Specification and Verification of a Real-Time Priority Queue," Technical Report
73. Yodaiken, V. and K. Ramamritham, "Mathematical models of real-time scheduling," (to appear) in Foundations of Real-Time Computing: Formal Specifications, edited by Andre van Tilborg, Kluwer Academic Publishers.

74. Yodaiken, V. and K. Ramamritham, "Specifying and Verifying a Real-Time Priority Queue with Modal Algebra," *Proc Real-Time System Symposium*, Dec. 1990.
75. Zhao, W. and J. A. Stankovic, "Performance Analysis of FCFS and Improved FCFS Scheduling Algorithms for Dynamic Real-Time Computer Systems," *Real-Time Systems Symposium*, pp. 156-165, Dec. 1989.
76. Zhao, W., J. A. Stankovic and K. Ramamritham, "A Window Protocol for Transmission of Time Constrained Messages," *IEEE Transactions on Computers*, Vol. 39, No. 9, Sept. 1990, pp. 1186-1203. Also COINS Technical Report 87-110.
77. Zhao, W., J. A. Stankovic and K. Ramamritham, "A Multi-Access Window Protocol for Transmission of Time Constrained Messages," *8th International Conf. on Distributed Computer Systems*, June 1988.

PI Names: Krithi Ramamritham, John A. Stankovic, Victor Yodaiken
PI Institution: University of Massachusetts
PI Phone Number: 413 545-0196 or 413 545-0720
PI E-mail Address: krithi@nirvan.cs.umass.edu, stankovic@cs.umass.edu
Grant or Contract Title: Dependable Real-Time Systems
Grant or Contract Number: N00014-85-k-0398
Reporting Period: 1 Oct 87 - 30 Sep 91

4.2 Presentations, Etc.:

1. Panelist (Ramamritham):

IFAC Workshop on Distributed Computer Control Systems, Sep 1991.
Conference on Very Large Databases, Barcelona, Spain, Sep 91. (chair)
9th and 10th International Conference on Distributed Computing Systems, June 1989 and 1990.
Workshop on Real-Time Operating Systems, May 1989; ACM SIGMOD, June 1989;
9th International Conference on Distributed Computer Systems, June 1989

2. Panelist (Stankovic):

8th IEEE Workshop on Real-Time Operating Systems and Software May 1991.
11th DCS Conference, May 23, 1991.
ONR Kickoff meeting, Washington, DC, November 8, 1990.
Workshop on Operating Systems for Mission Critical Computing, Sept. 19-21, 1989
Real-Time Operating Systems Workshop, Charlottesville, VA., May 10, 1990
NSF External Review Panel for initiation grants
COINS Industrial Affiliates meeting, May 4-5, 1990.
Workshop on Real-Time Operating Systems, (2 panels), May 1989; EuroMicro Workshop on Real-Time, Como, Italy, June 1989.

3. Conference Committees, etc. (Ramamritham):

Program Chair:

IEEE Real-Time Operating Systems Workshop, Boston, July 1977.

Program Co-Chair:

8th IEEE Workshop on Real-Time Operating Systems and Software May 1991.

Member, Conference Program Committee:

Second International Workshop On Responsive Computer Systems, Oct 1992.

IFAC Workshop on Distributed Computer Control Systems, August 1992.

IFAC/IFIP Workshop on Real-Time Programming, June 1992.
 ACM SIGMOD International Conference on Management of Data, May 1992.
 International Workshop On Configurable Distributed Systems, March 1992.
 International Conference on Data Engineering, Feb. 1992.
 Conference on Parallel and Distributed Information Systems, Dec 1991.
 Real-time Systems Symposium, Dec 1991.
 IFAC Workshop on Distributed Computer Control Systems, Sep 1991.
 International Conference on Distributed Computing Systems, May 1991.
 Real-time Systems Symposium, Dec 1990.
 NSF Site Visit Panel, June 1991.
 Program Co-Chair: 8th IEEE Workshop on Real-Time Operating Systems and Software and Workshop on Transactions and Objects, OOPSLA/ECOOP, 1990.
 Member, Conference Program Committee: 1990 Real-Time Systems Symposium; 7th Workshop on Real-Time Operating Systems and Software, May 1990; 10th International Conference on Distributed Computing Systems, May-June 1990; 1989 Real-Time Systems Symposium.
 Conference Session Chair: 10th International Conference on Distributed Computing Systems; 1989 Real-Time Systems Symposium.
 Program Committee, 10th International Conference on Distributed Computing Systems, 1990; Program Committee, 1989 Real-Time Systems Symposium; Program Committee, 9th International Conference on Distributed Computing Systems, 1989; Program Committee, 5th International Conference on Data Engineering, 1989. Member of proposal review panel for NSF Institutional Infrastructure Program (II), May 1989 and NSF Institutional Infrastructure Program, site visitor, July 1989. Member, Conference Program Committee (in progress): 10th IFAC Workshop on Distributed Computer Control Systems, Sept. 1991; 11th International Conference on Distributed Computing Systems, June 1991.
 Member, Program Committee, 8th DCS, 4th and 5th International Conference on Data Engineering.

Conference Session Chair:

IFAC Workshop on Distributed Computer Control Systems, Sep 1991.

4. Conference Committees, etc. (Stankovic):

Program Committee, Ninth Real-Time OS Workshop, Pittsburgh, May 1992.
 Organizing and Program Committee, Workshop on Artificial Intelligence in Real-Time, April 1991.
 Program Committee, Real-Time Systems Symposium, December 1991.
 Program Committee, ACM Sigmetrics, May 1992.
 Program Committee, AIDA '90, November 1990.
 NSF Research Initiation Grants, Databases, April 1991.
 IEEE Steering Committee for Distributed Computing.

Vice Chair for IEEE Real-Time Systems Technical Committee, January, 1991-present.

5. Conference Presentations (Ramamritham):

IFAC Workshop on Distributed Computer Control Systems, Sep 1991.

Conference on Very Large Databases, Barcelona, Spain, Sep 1991.

ACM SIGMOD, Denver CO, May 1991.

Program Co-Chair: International Conference on Distributed Computing systems, June 1990, Paris, France.

Program Committee: AIDA '90, November 1990; 3rd Workshop on large Grain Parallelism, Oct. 1989.

IEEE Steering Committee for Distributed Computing. International Conference on Distributed Computing Systems, June 1990, Paris, France

3rd Workshop on Large Grain parallelism, Oct. 1989.

Program Co-Chair, International Conference on Distributed Computing Systems, to be held June 1990, Paris, France; Member of Advisory Council on Programming of Parallel and Distributed Computers, Dept. of Computer Science, University of Texas, Austin, TX; Program Committee, 3rd Workshop on Large-Grain Parallelism, October 1989; NSF Research Initiation Grants Panel, Software Systems, April 1989; Organizing Committee, Workshop on RTAI, part of IJCAI, August 1989; Program Committee, Real-Time OS Workshop, CMU, May 1989; Program and Awards Committee, 9th International Conference on Distributed Computing systems, 1988; Program Committee, AIA Workshop, CMU/SEI, 1988-89; Program Committee, Real-Time Systems Symposium, Huntsville, AL, 1988; Program Committee, 15th International Symposium on Computer Architecture, 1988. Tutorial Speaker at the 1990 International Conference on Distributed Computing Systems (with Prof. Hermann Kopetz of Technical University, Vienna), a one-day tutorial on "Real-time Systems and Fault Tolerance."

Conference on Parallel Processing Chicago, IL, August 1989.

9th International Conference on Distributed Computing Systems, Newport Beach, CA, June 1989.

SIGMOD, May-June 1989; Sixth Workshop on Real-Time Operating Systems, Pittsburgh, PA, May 1989

1988 Real-time Systems Symposium, December 1988.

7th DEC conference, Berlin, September 1987.

6. Conference Presentations (Stankovic):

Conference on Very Large Databases, Barcelona, Spain, Sep 1991.

Keynote Speaker: Real-Time ADA and Embedded Systems Workshop, Nov. 1989

Keynote Speaker: Process Control Systems '91, Berlin, Feb. 1991

Real-Time ADA and Embedded Systems Workshop, Nov. 1989.

EuroMicro Workshop, Denmark, June 1990.

Invited speaker, ARTEWG meeting, four-hour presentation of Spring Project results, May 1990.

ONR Workshop, Annual Review, Oct. 1989

8th DEC Conference, San Jose, CA June 1988
Real-Time Systems Symposium, December 1987
Large Grain parallelism Workshop, October 1987
Real-Time OS Workshop, July 1987
General Chairman, Real-Time Operating systems Workshop, Washington, DC, May 1988.
IEEE CS Speaker, Distinguished Visitor Program, 1987-88.

7. Other Presentations (Ramamritham):

York University, York, U.K., Sep 1991.
IEEE Chapter, Bangalore, India, August 1991.
Oregon Graduate Institute, OR, January, 1991.
University of Pennsylvania, PA, October, 1990.
ONR Annual Review, research summary presentation, Wash., DC, Oct 1990.
Talks given at: Bellcore, Morristown, NJ, July 1990; General Electric Research Labs. Schenectady, NY, June 1990; Old Dominion University, Norfolk, VA Oct. 1989; Technologies Research Center, Hartford, CT, September 1989; AT&T Bell Laboratories, Murray Hill, NJ, August 1989; IBM Research Laboratories, Yorktown Heights, NY, August 1989; Jet Propulsion Laboratory, Pasadena, CA, December 1988; NASA, Ames Research Center, December 1988; and Pennsylvania State University, State College, PA, November 1988.
University of Newcastle Upon Tyne, UK, December 1987 and January 1988;
University of Warwick, Coventry, UK, February 1988.
Technical University, Eindhoven, The Netherlands, March 1988.
IEEE Workshop on Dextrous Robot Hands, April 1988; Technical University, Vienna, Austria, June 1988.
1988 EURO-TIMS Joint International Conference, Paris, France, July.

8. Other Presentations (Stankovic):

Invited Speaker, Modcomp Professional Seminar Series, Boston, June 7, 1991.
Keynote Speaker, Process Control Systems '91, Feb. 1991 (unable to attend).
Invited Speaker, Tutorial, Hard Real-Time Systems, Berlin, Feb. 1991.
ONR Annual Review, research summary presentation, Wash., DC, Oct. 25, 1990.
Dept. of Computer and Info. Science, Technology Roundtable, presentation, Boston, Oct. 23, 1990.
As part of CRICCS preparation, traveled to California and gave three presentations at Stanford, Rockwell and FMC and visited JPL.
Presented seminars: University of Groningen, The Netherlands, June 1990; Draper Labs, April 1990; Universite Pierre and Marie Curie, Paris, France, January 1990.
Ready Systems, Seminar, Sunnyvale, CA, July 1989.

Naval Postgraduate School, Distinguished Lecture Series, July 1989.
 Charles Stark Draper Labs, Seminar, May 1989.
 Northeastern University, Distinguished Lecture Series, May 1989.
 Purdue University, CS Dept, Seminar, February 1989.
 University of Michigan, Distinguished Lecture Series, February 1989.
 University of Maryland, Seminar, November 1989.
 Texas Instruments, Seminar program, Dallas, August 1988
 8th DCS Conf. panel, San Jose, CA, June 1988
 Panel on Real-Time Computing, RTOS Workshop, Washington, DC, May 1988
 University of Maryland, CS Seminar, March 1988
 University of Connecticut, CS Seminar, November 1987
 Panel on Critical Issues in Distributed Operating Systems, DCS Conf., Berlin, September 1987.

9. Board Memberships, etc. (Ramamritham):

Associate Editor, *International Journal of Real-Time Systems*, 1989-present.
 AT&T Bell Laboratories, Murray Hill, NJ, consultant, July-August 1991.
 Visiting Fellowship, Science and Engineering Research Council, UK September 1987 - June 1988.

10. Board Memberships, etc. (Stankovic):

ACSIOM Board of Directors, 1989-present.
 Co-Editor-in-Chief and Co-Founder, *International Journal of Real-Time Systems*.
 Book Series Editor, Research Monographs in Real-Time Systems, Kluwer Academic Publishers.
 Editor, *IEEE Transactions on Computers*, July 1986-Dec. 1990.
 Ready Systems, Senior Scientific Advisory Board, current.
 Advanced System Technologies, Senior Technical Advisory Board, current.

11. PhD Students - Thesis Chairman (Ramamritham):

P. Chrysanthis, ACTA, A Framework for modeling and reasoning about Extended Transactions, July 1991. K. Arvind, (Co-Chair with Stankovic), Protocols for Distributed Real-Time Systems, May 1991.

12. PhD Students - Thesis Chairman (Stankovic):

Jim Huang, Real-Time Transaction Processing: Design, Implementation and Performance Evaluation, May 1991
 K. Arvind, (Co-Chair with Ramamritham), Protocols for Distributed Real-Time Sys-

tems, May 1991.

13. Awards (Stankovic):

IEEE Computer Society's Meritorious Service Award, 1991

International Advisory Committee, *Journal of Computer Science and Informatics*,
Computer Society of India, May 1990.

Full Professor, Dept. of Computer and Information Science, September 1, 1990.

PI Names: Krithi Ramamritham, John A. Stankovic, Victor Yodaiken
PI Institution: University of Massachusetts
PI Phone Number: 413 545-0196 or 413 545-0720
PI E-mail Address: krithi@nirvan.cs.umass.edu, stankovic@cs.umass.edu
Grant or Contract Title: Dependable Real-Time Systems
Grant or Contract Number: N00014-85-k-0398
Reporting Period: 1 Oct 89 - 30 Sep 90

5. Research Transitions and DoD Interactions:

Indirect Research Transitions:

Stankovic has been working with Applied Technology Systems for The Naval Surface Warfare Center, White Oak Lab on integrating real-time scheduling results from Spring and other research groups into a Knowledge Based DOS Assistant.

Ramamritham has been collaborating with the researchers at AT&T Bell Labs, Murray Hill and has completed extending Concurrent C with constructs for specifying real-time constraints. These incorporate the notion of guarantee as well.

We are also working on a new tutorial text entitled, Advances in Real-Time Systems. The text has been approved by the IEEE.

The Spring project formed one of the focal points of the department's Science and Technology Proposal to NSF. It was to provide the necessary systems support for complex real-time systems. As part of this proposal, interactions with several industrial participants have been initiated. One of these was Texas Instruments. Some of their real-time work has been influenced by Spring scheduling algorithms and we continue to work with them.

PI Names: Krithi Ramamritham, John A. Stankovic, Victor Yodaiken
PI Institution: University of Massachusetts
PI Phone Number: 413 545-0196 or 413 545-0720
PI E-mail Address: krithi@nirvan.cs.umass.edu, stankovic@cs.umass.edu
Grant or Contract Title: Dependable Real-Time Systems
Grant or Contract Number: N00014-85-k-0398
Reporting Period: 1 Oct 90 - 30 Sep 91

6. Description of Software and Hardware Prototypes:

We further developed the software simulation testbed for distributed real-time systems. Documentation has been completed, and the new graphics front-end which animates the scheduling algorithms we develop has also been developed for this testbed. We further enhanced our Software Generation System which enables the specification of task requirements and the subsequent creation of the Spring Kernel with this information embedded in the run time data structures. Subsequent enhancements are being planned for this tool in order to handle task groups.

RT-CARAT, a distributed database testbed, has many real-time transaction protocols implemented and running. Further enhancements, such as real-time buffer management, incorporation of abstract data type objects, and provision for structured transactions are in progress.

A prototype of Real-Time Concurrent C is nearing completion and should be available for distribution within the next few months.

Development of the distributed version of the Spring Kernel is currently in progress using Systran's SCRAMNet, a replicated shared memory architecture that uses a fiber optic register insertion ring, running at 150 MBits/sec. Each node has 2 MB of shared common memory, and writes to this memory are broadcast (circulated) about the ring. We are currently implementing the IPC primitives with SCRAMNet. SCRAMNet presents some attractive features for implementing distributed scheduling and for realizing fault tolerance. These are currently under investigation.